

**WHAT IS CLAIMED:**

1. A method of deploying software across an enterprise comprising:  
 registering a set of users of an enterprise:  
 at a deployment master workstation, receiving an administrator request to scan registered users to  
 5 determine users requiring a deployment;  
 receiving from a process running in said user machines data indicating configuration status;  
 at said user machines, initiating a native setup application, with execution parameters received  
 from said deployment master;  
 at said user machines, confirming said deployment.
- 10 2. The method of claim 1 further comprising;  
 at one or more user machines executing a configuration detection application that outputs a  
 structured file that is easily imported into a remote database.
3. The method of claim 1 wherein said application gathers configuration parameters and parses  
 and output said parameters in a form that can be accessed by a central database.
- 15 4. The method of claim 3 wherein said parameters comprise values of BIOS/PCI, PnP, or other  
 registers.
5. The method of claim 3 wherein said output comprises a structured ASCII file that can be  
 natively input by a bulk database.
- 20 6. A method of deploying software across an enterprise comprising:  
 a one-time application that runs and collects configuration data into a flat file anywhere or in a  
 logon server;  
 to database;  
 administrator picks a software configuration for a machine, engine generates a parameter file.  
 database generates a receiving from agents installed in said user machines data indicating  
 25 configuration status;  
 at said user machines, initiating a native setup application, with user inputs received from said  
 deployment master;  
 at said user machines, confirming said deployment.
7. The method of claim 1 wherein said software comprises:  
 30 a WindowsNT operating system.

8. The method of claim 1 wherein said software comprises:  
a Windows2000 operating system.

9. The method of claim 1 wherein said software comprises:  
a Windows operating system subsequent to Windows2000.

10. The method of claim 1 wherein said deployment integrates with Electronic Software Distribution (ESD) packages.

11. The method of claim 1 further comprising:  
storing deployment history in a database.

12. The method of claim 1 wherein said deployment may proceed via Multicast from a Preboot Execution Environment (PXE).

13. The method of claim 1 further comprising controlling bandwidth over a deployment media by:

- measuring bandwidth capacity over a deployment path;
- calculating total data transfer for a particular deployment session;
- controlling deployment transfer bandwidth to minimize QoS degradation while meeting maximum allowed deployment latency.

14. The method of claim 1 wherein said software includes a security identification component that must be installed in native mode.

15. A system for automatically deploying software comprising:  
a first tier comprising a PXE and NT authentication component and a deployment services server;  
a second tier comprising SSL, Certificate and LDAP authentication communication to an intranet connecting to said first tier;  
a third tier comprising SQL and NT authentication;  
a fourth tier comprising MTS transaction integrity; and  
a fifth tier comprising an SQL database and RDBMS Referential integrity and encryption.

16. A system for automatically deploying software comprising:  
a scan process that collects detailed hardware configurations that are loaded in a relational database for the purpose of relating hardware configurations to operating system software configurations to determine which, if any, hardware is compatible with versions and configurations of an operating system;

a script engine at a target machine that initiates processes based on a batch input that is exported from a relational database at a deployment server;

a Visual Driver Manager that passes hardware peripheral, component and ASIC specific information to a script engine from a database for the purpose of generating a specific operating system configuration that will install a specific set of drivers specific to the target machine's hardware configuration at the time the OS installation or upgrade occurs;

a Visual OS Manager Custom generation of Operating System installation or upgrade source trees that consolidate redundant files and identifies version and language unique files to permit multiple versions, languages and licensing models of an operating system to be installed from a single, unified source tree;

a Visual Package Manager that moves a native or ESD repacked software application installation package from a source file system to an operating system installation or upgrade source tree for the purpose of pushing a native or ESD installation package to a target during OS installation or upgrade;

a relational database that contains hardware and software configuration attributes specifically related for the purpose of generating operating system configuration parameters as xml messages or structured ascii text output files that can be interpreted by element 1 for the purpose of repeatable, custom operating system installations;

17. The system of claim 16 further comprising:

a client server application with a browser based GUI that permits a user to select microcomputer(s) and apply an operating system configuration ad-hoc or previously defined as a software configuration template.

18. The system of claim 16 further comprising:

an agent service that packages operating system configuration files and operating system files and executables for launching a specific deployment type and media and tracks deployment states using store and forward messaging to indicate operating system installation or upgrade progress milestones.

19. The system of claim 16 further comprising:

a Deployment Rollback manager that is Triggered during an operating system upgrade and uses an operating system client executed by PXE or floppy to restore operating system files required to successfully "restore" an operating system that existed prior to current installation or upgrade of OS.

20. An electronic data file, recorded or transmitted on a digital medium, that when loaded into an appropriately configured digital apparatus causes the apparatus to operate in accordance with the method of claim 1.

21. An electronic data file, recorded or transmitted on a digital medium, that when loaded into an appropriately configured digital apparatus causes the apparatus to embody the system of claim 15.

22. A method for executing a deployment at a target machine comprising:  
utilizing a script engine to read input parameters defined as name value pairs, said input parameters generated by a server database application; and  
at said script engine, initiating instructions causing installation or upgrade of a software system.

23. The method of claim 22 wherein said name value pairs may be customized by a user through a browser or client application interface.

24. The method of claim 22 wherein said installation or upgrade further comprises:  
updating software preexisting on said target machine.

25. The method of claim 22 further comprising:  
relating a specific hardware configuration to a specific software configuration in a relational database in a server;  
generating parameterized files consisting of name/value pairs;  
reading said parameterized files by a script engine;  
setting environment variables or varibilized 1parameters; and  
building an installation file to be read by installation routines at a target machine.

26. A method of preparing for deployment of operating system or application software comprising:

collecting one or more detailed hardware configurations from target machines;  
loading said configurations into a relational database; and

using said database for relating hardware configurations to operating system or application software configurations to determine which, if any, hardware is compatible with versions and configurations of an operating system.

27. The method of claim 26 wherein said collecting comprises at a target machine scanning configuration registers for the purpose of creating a machine specific hardware configuration in a structured file that can be bulk copied or read natively by a relational database.

28. The method of claim 26 wherein said collecting comprises retrieving vendor id, product id, product description, subsystem id and subsystem class for use as data keys in a relational database.

29. The method of claim 26 wherein said configuration registers includes one or more of PCI, PnP, SMB, DMI, USB, PCMCIA, CardBus, IEEE 1394, ATA, SCSI and Video BIOS registers.

30. The method of claim 27 further comprising:  
parsing and interpret said configuration registers; and  
formatting output that can be natively bulk copied into a relational database.

31. The method of claim 30 further comprising:  
wherein said output is an xml message.

32. The method of claim 30 further comprising:  
wherein said output is a structured ascii file.

33. A method of preparing for deployment of operating system or application software comprising:

collecting one or more detailed hardware configurations at target machines;  
loading said configurations into a relational database; and  
using said database for relating hardware configurations to operating system or application software configurations to determine which, if any, hardware is compatible with versions and configurations of an operating system.

34. The method of claim 33 further comprising:  
adding driver support to an OS source tree by parsing third party driver OS information files to identify the binaries, installation definitions and adding them to an operating system source tree.

35. The method of claim 34 further comprising:  
tracking specific drivers, versions, and compatibilities from said source tree in a source tree in said database.

36. The method of claim 34 further comprising:  
using a graphical user interface, on can assert a source device, whereby the drivers are uploaded to a specific directory that can be tracked in said database.

37. The method of claim 33 further comprising:

optionally editing the original installation file and merging it with the source tree in such a way as to enable the operating systems hardware detection facility to install the device by automatic detection or by explicitly defining the hardware in an information file that the operating system reads during OS installation.

5 38. The method of claim 33 further comprising:

retrieving vendor id, product id, product description, subsystem id and subsystem class for use as data keys in a relational database, said data keys used as parameters for installation flags for one or more system components requiring direct, version specific, hardware driver support.

10 39. A method of custom generation of operating system (OS) installation source trees comprising:

consolidating redundant files;

identifying version and language unique files thus permitting multiple versions, languages and licensing models of an operating system to be installed from a single, unified source tree;

15 retrieving CD-ROM volume and serial number and create database records of OS version, type & language;

wherein said information is retrieved from a form a user completes that is posted to a database table;

wherein a primary key in this table will be a PID read from a CD;

providing an interface allowing an end user to override detected Operating System and Language;

20 prompting a user to select the appropriate Operating system license model;

prompting a user to delete uploaded CDs;

prompting a user if there is an attempt to upload same CD-ROM more than once;

assigning each operating system a unique ID code, said code be used to determine the OS combinations.

25 40. The method of claim 39 further comprising:

moving a native or ESD repacked software application installation package from a source file system to an operating system installation source tree for pushing a native or ESD installation package to a target during OS installation;

30 using generic "forms" in the front end and tables in the database that permit a user to enter the name, source path and installation parameters of a software installation package or ESD client.

writing OS and ESD configuration to parameter files for passing the OS configuration and “ESD packages” to be installed on a target machine to a script engine which will then copy a custom “set” of packages per installation.

41. The method of claim 39 further comprising:

5 providing a multi-cast publishing services to multi-cast client subscribers; wherein the agent interacts with the database for authentication of subscribers and configuration of file sets of subscriber sessions; and

during OS installation, a “multi-cast” client is installed and optionally executed on the target PC that “reads” the installation parameter file and subscribes to a file based multi-cast session after  
10 the operating system is running in GUI mode and has authenticated to the Deployment Agent.

42. A method of operating a deployment database holding hardware and software configuration attributes specifically related for the purpose of generating operating system configuration parameters for the purpose of repeatable, custom operating system installations comprising:

create database tables to represent entities for managing deployment;

15 creating database tables to represent parameters and values of an operating systems installation facility to include elements of an operating system source tree required to perform an installation of a specific language, version, and license model;

creating database tables to represent operating system and application license keys per configuration;

20 relating one or many hardware configurations to one or many operating system configurations;

relating processes to customers by demographics and network directories so that elements can be selected using customers, sites, departments and networks as operating system configuration selection criterion.

43. The method of claim 42 further comprising:

25 including a data import facility to import files created by scanning in target machines.

44. The method of claim 42 further comprising:

including a data export facility to export parameter files or messages for a script engine at a target machine.

45. A client/server application with a browser based GUI that permits a user to select  
30 microcomputer(s) and apply an operating system configuration comprising:

a secure client interface that accesses a database for the purpose of providing a simple to use interface that enables a non-technical user to create complex, repeatable operating system configurations for deployment to target machines;

an administrative console that allows a network administrator to demographically model the deployment database to resemble the domain or directory objects;

a hardware configuration form that allows a user to view the target machines hardware configuration, OS compatibility and optionally choose what components are to have drivers support installed.

a hardware configuration form that allows a user to designate a machine or components of a machine as compatible with a specific OS version and patch level.

a software configuration form that allows a user to define and choose and configure an operating system installation by language, license model, version and patch level.

a software and system configuration management form that allows a user to generically define packages or utilities to be pushed to a target machine during operating system installation. and

a form that permits users to move machines between customers, sites and departments to reflect physical moves.

46. The application according to claim 45 further comprising:  
wherein said domain or directory objects comprise customers, sites, departments and deployment servers.

47. The application according to claim 45 further comprising:  
an interface that allows ASP and customers to collaborate on deployment projects.

48. The application according to claim 45 further comprising:  
user id deployment roles that constrain user privileges by demographics and administrative capabilities.

49. The application according to claim 45 further comprising:  
a multi-language browser translation facility that permits users to view the application in their preferred language.

50. The application according to claim 45 further comprising:  
wherein said browser application is an N-Tier DNA application.



51. A deployment manager agent service that packages operating system configuration files and operating system files and executables for launching a specific deployment type and media wherein said agent service tracks deployment states using store and forward messaging to indicate operating system installation progress milestones and further comprising:

5 a PXE that delivers and executes operating system deployments using a pre-boot execution environment for specific target machines by identifying hardware elements which represent data keys for parameter files delivered to a PXE server by a service agent.

52. The agent service according to claim 51 further wherein a 16bit OS delivers and executes operating system deployments using a 16 bit operating system client for specific target machines by  
10 identifying hardware elements which represent data keys for parameter files delivered to a bootable media by a service agent.

53. The agent service according to claim 51 further wherein a 32 bit OS delivers and executes operating system deployments using a 32 bit operating system client for specific target machines by  
15 identifying hardware elements which represent data keys for parameter files delivered to a bootable media by a service agent.

54. The agent service according to claim 51 further wherein a 64 bit OS delivers and executes operating system deployments using a 64bit operating system client for specific target machines by  
20 identifying hardware elements which represent data keys for parameter files delivered to a bootable media by a service agent.

55. The agent service according to claim 51 further comprising a Wake On Lan triggering operating system deployments using a wake on Lan calls for specific target machines by identifying hardware elements which represent data keys for parameter scripts delivered to a bootable media by a service agent.

56. The agent service according to claim 51 further comprising scheduling a cron-like facility  
25 that submits deployment configurations based on a user defined calendar.

57. A method of providing for a deployment rollback comprising:  
triggering, during an operating system upgrade, a deployment rollback process;  
said process using an operating system client executed by PXE or floppy to restore operating  
system files required to successfully "restore" the 16, 32 or 64 bit operating system that existed  
30 prior to installation of a new operating system;

verifying that disk and partition space meet rollback requirements for each target PC.  
 performing backup on a system root directory of the operating system targeted for upgrade storing  
 the files on a local or network file system;  
 performing backup on a registry of the operating system targeted for upgrade storing the files on a  
 5 local or network file system;  
 performing backup on an OS installation source directory (if present) of the operating system  
 targeted for upgrade storing the files on a local or network file system;  
 performing backup on a DM install directory (if present) of the operating system targeted for  
 upgrade storing the files on a local or network file system;  
 10 performing backup on a Windows OS directory of the operating system targeted for upgrade  
 storing the files on a local or network file system;  
 performing backup on a MBR, boot sector and file allocation tables of the of the operating system  
 targeted for upgrade storing the files on a local or network file system; and  
 performing backup on user specified files and/or directories of the operating system targeted for  
 15 upgrade restore.

58. The method of claim 57 further comprising:

restoring the Windows OS directory of the operating system targeted for upgrade reading the  
 specific target's files from a local or network file system;  
 restoring the registry of the operating system targeted for upgrade reading the specific target's  
 20 files from a local or network file system;  
 restoring the system root directory of the operating system targeted for upgrade reading the  
 specific target's files from a local or network file system;  
 restoring the OS installation source directory (if present) of the operating system targeted for  
 upgrade reading the specific target's storing the files on a local or network file system;  
 25 restoring MBR, boot sector and file allocation tables of the of the operating system targeted for  
 upgrade storing the files on a local or network file system;  
 restoring the DM install directory (if present) of the operating system targeted for upgrade reading  
 specific target's files from a local or network file system; and  
 restoring backup on user specified files and/or directories of the operating system targeted for  
 30 upgrade restore.